

Unbiased Bayesian Inference for Population Markov Jump Processes via Random Truncations

Anastasis Georgoulas¹, Jane Hillston¹, and Guido Sanguinetti^{1,2}

¹School of Informatics, University of Edinburgh

²SynthSys — Synthetic and Systems Biology, University of Edinburgh

We consider continuous time Markovian processes where populations of individual agents interact stochastically according to kinetic rules. Despite the increasing prominence of such models in fields ranging from biology to smart cities, Bayesian inference for such systems remains challenging, as these are continuous time, discrete state systems with potentially infinite state-space. Here we propose a novel efficient algorithm for joint state / parameter posterior sampling in population Markov Jump processes. We introduce a class of pseudo-marginal sampling algorithms based on a random truncation method which enables a principled treatment of infinite state spaces. Extensive evaluation on a number of benchmark models shows that this approach achieves considerable savings compared to state of the art methods, retaining accuracy and fast convergence. We also present results on a synthetic biology data set showing the potential for practical usefulness of our work.

1 Introduction

Discrete state, continuous time stochastic processes such as Markov Jump Processes (MJP) [1] are popular mathematical models used in a wide variety of scientific and technological domains, ranging from systems biology to computer networks. Of particular relevance in many applications are models where the state-space is organised according to a population structure (population Markov Jump Processes, pMJP): each state label corresponds to counts of individual entities in a number of populations (or species). These models are at the root of essentially all agent-based models, a class of models which is gaining increasing popularity in applications ranging from smart cities, to epidemiology, to systems biology. Despite their importance, solving inferential problems within the pMJP framework is challenging: the discrete nature of the system prevents the use of simple parametric distributions, and the size of the state space (which can be unbounded for open systems) effectively rules out analytical computations. At the same time, technological advances in areas as diverse as single cell biology and remote sensing are providing increasing amounts of data which can be naturally modelled as pMJPs, creating a pressing need for inferential methodologies.

In response to these developments, researchers in the statistics, machine learning and systems biology communities have been addressing inverse problems for MJPs using a variety of methods, from variational techniques [2, 3] to particle-based [4, 5] and auxiliary variable sampling methods [6]. Markov-chain Monte Carlo (MCMC) methods, in particular, offer a promising direction: while often computationally more intensive than variational methods, they provide asymptotically exact inference. However, standard MCMC methods rely on likelihood computations, which are computationally or mathematically infeasible for pMJPs with a large or unbounded number of states. Such systems are commonplace in many applications, where one is often confronted with open systems where upper bounds on the numbers of agents are difficult to come by. As far as we are aware, current methods address this issue by arbitrarily truncating the state space according to pre-defined heuristics, offering no control over the error introduced by this procedure.

In this paper we present a novel Bayesian approach to posterior inference in pMJPs which solves these issues by adopting a pseudo-marginal approach based on random truncations, yielding both asymptotic exactness and computational improvements. We build on the auxiliary variable Gibbs sampler for finite state Markov Jump Processes (MJP) of [6], significantly increasing its efficiency by leveraging the more compact representation of the kinetic parameters provided by the pMJP framework. We then present a novel formulation of the likelihood, which enables the deployment of a *Russian Roulette*-like random truncation strategy as in [7, 8]. Based on this, we develop a pseudo-marginal sampling approach for general pMJPs, obtaining two novel algorithms: a relatively straightforward Metropolis-Hastings pseudo-marginal scheme, and an auxiliary variable pseudo-marginal Gibbs sampler. We examine the performance of these algorithms in terms of accuracy and efficiency on non-trivial case studies. We conclude the paper with a discussion of our contribution in the light of existing research and possible future directions in systems biology.

2 Background

2.1 Population Markov Jump Processes

Population Markov Jump Processes are a particular type of Markov Jump Processes (also known as Population Continuous Time Markov Chains); they are continuous time stochastic processes whose discrete state vector $\mathbf{s} = (n_1, n_2, \dots, n_M)$ gives the agent counts of each of M populations (or species) interacting through R reaction channels. We will adopt here the language of chemical reactions to describe such processes, but the same considerations apply in general. Reactions between individual agents (or molecules) happen as a result of random collisions, and each reaction changes the state by a finite amount, encoded in the *stoichiometry* of the system, corresponding to the creation/ destruction of a certain number of molecules. Each reaction i also has an associated *kinetic law* giving its rate: this is generally of the form

$$f_i(\mathbf{n}) = \theta_i \rho_i(\mathbf{n}), \quad (1)$$

where ρ_i is a *fixed* function of the state \mathbf{n} , while θ_i are (usually unknown) kinetic parameters. Therefore, while in a general MJP there can be a parameter associated with each possible transition, in pMJPs the dynamics are captured more succinctly by a single parameter per reaction. A schematic of a simple pMJP is given in Figure 1, where it can be seen that the same reaction can correspond to multiple transitions in the state-space of the process, all of which follow the same kinetic law and incur the same update to the state.

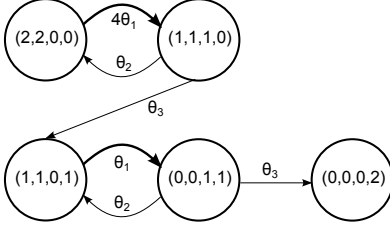


Figure 1: State-space of an example system. Arrows indicate transitions between states; the bolded transitions are “instances” of the same reaction type, which updates the state by $(-1, -1, 1, 0)$ and occurs with rate $\theta_1 s_1 s_2$, where s_1, s_2 are the first and second components of the state.

The time evolution of the process marginals is given by the Chemical Master Equation (CME):

$$\frac{dp_i(t)}{dt} = \sum_{j \neq i} p_j(t) a_{ji} - p_i(t) \sum_{j \neq i} a_{ij} \quad (2)$$

where $p_i(t)$ is the probability of being in state i at time t and a_{ij} is the rate of jumping from state i to state j , which for pMJPs is known from the kinetic law.

For finite state-spaces, one can gather the transition rates a_{ij} in the *generator matrix* \mathbf{A} , and the CME can be solved analytically as:

$$\mathbf{p}(t) = \mathbf{p}(0)e^{\mathbf{A}t} \quad (3)$$

This solution can be computationally intensive, even with the use of specialized algorithms like [9].

2.2 Uniformisation and inference

An alternative approach to solve the CME is given by *uniformisation* [10], a well-known technique for the transient analysis of Markovian systems, used widely in fields like performance modelling. Given a MJP with generator \mathbf{A} , uniformisation constructs a discrete time Markov chain by imposing a common exit rate γ for all states. For this procedure to be consistent, γ must be no less than the highest exit rate among all states. The resulting *uniformised* system is then faster than the original, in the sense that transitions occur at a higher rate. To compensate for this and maintain the behaviour of the original MJP, virtual jumps must be added from each state to itself. This results in a discrete time system with transition probability matrix $B = \frac{1}{\gamma}\mathbf{A} + I$, in which likelihood computations are standard. In this discrete time system, the waiting time before a jump occurs now follows an exponential distribution with rate γ , regardless of the current state. The probability of jumping from state i to another state j is $\frac{a_{ij}}{\gamma}$, but it is now also possible to remain in i after the jump, with probability $1 - \frac{1}{\gamma} \sum_{j \neq i} a_{ij}$. Jensen’s classical result [10] then guarantees that all the time-marginals of the discrete time process match those of the continuous time chain.

Uniformisation has previously been exploited by [6] to draw posterior samples from a MJP conditioned on a set of observations. The idea is to construct a discrete time chain using uniformisation, sample a trajectory (including self-loops) and run a standard forward filtering-backward sampling (FFBS) algorithm on it. This gives a new trajectory which, when self-jumps are removed, is a sample from the posterior process. This path-sampling algorithm can be alternated with Gibbs updates to jointly sample transition probabilities; in [6] this is

accomplished by choosing conjugate Dirichlet priors on each entry of the generator matrix, resulting in potentially many parameters with consequent storage/ computational issues.

3 Unbiased sampling for pMJPs

3.1 Efficient Gibbs sampling for finite state pMJPs

The special structure of pMJP systems implies considerable inferential savings over the generic Gibbs sampler [6]. In particular, the functional form of the kinetic law associated with the i th reaction, $f_i(\mathbf{s}) = \theta_i \rho_i(\mathbf{s})$, suggests a different conjugate prior for the parameters θ_i , which greatly simplifies the parameter sampling steps within the Gibbs sampler.

Let (S, T) be a full trajectory sampled from the uniformised conditional posterior in a Gibbs step, where $S = (s_0, s_1, \dots, s_K)$ is the sequence of states at times $T = (t_0, t_1, \dots, t_K)$. Let u_k denote the reaction at time t_{k+1} , as inferred¹ from inspection of s_k and s_{k+1} . From Section 2.1, we know that the total rate of exiting state s_k is $r_k = \sum_{i=1}^R \theta_i \rho_i(s_k)$. Since the waiting time between jumps is exponentially distributed in a MJP, this gives

$$p(t_{k+1} \mid t_k, s_k) = r_k e^{-\Delta t_k r_k}, \text{ where } \Delta t_k = t_{k+1} - t_k$$

The probability of the next state being s_{k+1} is $\frac{\theta_{u_k} \rho_{u_k}(s_k)}{r_k}$. The total likelihood is then

$$\begin{aligned} L(\boldsymbol{\theta}) &= p(S, T \mid \boldsymbol{\theta}) = p(S \mid \boldsymbol{\theta}) p(T \mid S, \boldsymbol{\theta}) \\ &= \prod_{k=0}^{K-1} \frac{\theta_{u_k} \rho_{u_k}(s_k)}{r_k} r_k e^{-\Delta t_k r_k} \\ &= \prod_{k=0}^{K-1} \theta_{u_k} \rho_{u_k}(s_k) e^{-\Delta t_k r_k} \end{aligned}$$

Let each parameter be Gamma-distributed *a priori*:

$$p(\theta_i) = \frac{b_i^{a_i}}{\Gamma(a_i)} \theta_i^{a_i-1} e^{-b_i \theta_i}$$

We then have:

$$\begin{aligned} p(\theta_i \mid S, T) &\propto p(\theta_i) p(S, T \mid \boldsymbol{\theta}) \\ &\propto \theta_i^{a_i + N_i - 1} e^{-b_i - \sum_{k=0}^{K-1} \Delta t_k \rho_i(s_k) \theta_i} \end{aligned} \tag{4}$$

Therefore, conditioned on the trace, the parameters are again Gamma-distributed with shape $a_i + N_i$ and rate $b_i + \sum_{k=0}^{K-1} \Delta t_k \rho_i(s_k)$, where N_i is the number of times the i th reaction type is observed in the trace. Hence, we have *exact* Gibbs updates for the kinetic parameters; notice that, since we have a single parameter for each reaction, the number of parameters to be sampled is often orders of magnitude lower than the number of parameters sampled in [6] (one per possible state transition), yielding computational and storage savings.

¹We assume each reaction has a distinct update vector.

3.2 Unbounded state-spaces

Many pMJPs of practical interest describe open systems with infinite state-spaces, which are not amenable to uniformisation. A plausible solution would be to truncate the system, possibly using methods such as in [11] to quantify the error. However, any such bound would be dependent on the unknown parameters, and in order to achieve acceptable performance we may need to still retain very large state spaces. An alternative approach may be to introduce random truncations in such a way as to obtain an unbiased estimator of the likelihood, which can be used in a pseudo-marginal MCMC scheme [12, 13]. We describe here two algorithms based on random truncations, a simple Metropolis-Hastings (M-H) sampler directly targeting the marginal likelihood, and a Metropolized auxiliary variable Gibbs sampler.

3.2.1 Expanding the likelihood

We start by describing a formulation of the likelihood in the pMJP setting as an infinite series. The basic idea is to decompose the space of process trajectories into a nested sum over subspaces of trajectories which differ by at most N from the observations. We can then define a generator matrix on each of these finite state-space systems and compute transient probabilities using (3). We now explicitly define the terms in this expansion of the likelihood. For simplicity, we focus on deriving the likelihood for a single, noiseless observation (t', s') in a one-dimensional process, assuming the state at time 0 is known to be $s \in \mathbb{N}$. Due to the Markovian nature of the process, the actual likelihood will be given by a product of such terms. If we write $s^u = \max(s, s')$, we have:

$$\begin{aligned} p(s' \mid s, \boldsymbol{\theta}) &= \sum_{N=0}^{\infty} p(s', \max(s_{0:t'} - s^u) = N \mid x, \boldsymbol{\theta}) \\ &\equiv \sum_{N=0}^{\infty} p^{(N)}(s', s) \end{aligned} \quad (5)$$

The notation $s_{0:t}$ indicates all values of the process in the time interval $[0, t]$ and is used here as follows: $\max(s_{0:t}) = N$ means that the maximum value of the process in the interval $[0, t]$ is N . Similarly, $\max(s_{0:t}) \leq N$ means that the process does not exceed the value N during $[0, t]$.

Note that the constraint on the maximum of $s_{0:t'} - s^u$ does not simply define a state-space, but constrains us to consider only those trajectories that actually achieve a “dispersal” of N . If we define

$$f^{(N)}(s', s) = p(s', \max(s_{0:t} - s^u) \leq N \mid x, \boldsymbol{\theta})$$

then each term of the series can be decomposed as:

$$p^{(N)}(s', s) = f^{(N)}(s', s) - f^{(N-1)}(s', s) \quad (6)$$

These sub-terms are now the transient probability for a finite state-space pMJP, and can be computed using Equation 3. Any number of them are computable but, naturally, the whole sum cannot be computed in finite time. It can, however, be estimated in an unbiased way.

3.2.2 Random truncations

Assume we wish to estimate an infinite sum

$$f = \sum_{N=0}^{\infty} f_N$$

where each term f_N is computable. One way of approximating the sum is to pick a single term f_k , where k is chosen from any discrete distribution with mass p_0, p_1, \dots . We can immediately see that $\hat{f} = \frac{f_k}{p_k}$ has expectation $E[\hat{f}] = \sum_{N=0}^{\infty} \frac{f_N}{p_N} p_N = f$ and is therefore an unbiased estimator of the infinite sum. An issue with this approach is that, depending on the choice of distribution p_i , the variance of \hat{f} might be very large, even infinite.

A reduced variance estimator can be obtained by approximating f with a partial sum up to order N , weighted appropriately. The number of terms is chosen randomly: at every term j , a random choice is made: there is a probability q_j of stopping the sum, otherwise we continue to form iteratively the partial sum $\hat{f} = \sum_{N=0}^j \frac{f_N}{p_N}$, where $p_N = \prod_{j=1}^{N-1} (1 - q_j)$. This scheme, imaginatively termed *Russian Roulette sampling* [7], can also be shown to yield an unbiased estimator of f .

3.2.3 Metropolis-Hastings sampling

Applying this random truncation strategy to the expansion in (5) produces an unbiased estimator. Such estimates can be obtained for every interval between successive observations; since they are independent, their product will be an unbiased estimate of the likelihood under all the observations. Note that each summand in (5) is a probability, and is therefore non-negative. Thus, we avoid the problems of possibly negative estimators; this positivity is important, as non-positive estimators may result in a large or infinite variance. It is worth remarking that the term for $N = 0$ corresponds to a space that includes the observations at both ends of the time interval, and hence will already include a significant contribution of probability mass towards the likelihood.

The same approach is easily extended to higher dimensions, where the states are vector of integers, by adapting the notation: $\max(\mathbf{s}_{0:t}) \leq N$ means that the value in *any* dimension does not exceed N in the given interval, whereas $\max(\mathbf{s}_{0:t}) = N$ now means that a value of N is not exceeded in any dimension during $[0, t]$, and that it is achieved in at least one dimension.

This procedure directly gives rise to a pseudo-marginal M-H algorithm, where the likelihood term is approximated by the unbiased estimate obtained as described above. We refer to this as *Algorithm 1* and examine its performance in the next section.

For our purposes, we choose a q_n sequence such that the probability of accepting a term decreases geometrically; specifically, we use $q_n = 1 - a(1 - q_{n-1})$, with $q_0 = 0$ and $a = 0.95$. We note that, since all terms in the series are non-negative and tend to 0, we can make use of a result from [7] to show that the variance of the estimator is finite. We show an empirical analysis of the variance in Section 4.1 that validates our choice of q_n and indicates that performance is robust with respect to the choice of the particular stopping distribution.

3.2.4 Modified Gibbs sampling

An alternative approach is to incorporate the truncation in the Gibbs sampler described in Section 3.1. The difficulty is that there is no direct way to sample trajectories without a bound on the state-space, as the uniformisation sampler requires a finite number of states. To work around this limitation, we propose to sample a truncation point, then draw a trajectory and parameters for this state-space as in Section 3.1. Since we are no longer sampling from the true conditional posterior over trajectories, but rather are also conditioning on the chosen truncation, we are no longer able to accept every trajectory and parameter sample drawn. Instead, we must introduce an acceptance ratio that will ensure we are sampling from an unbiased estimate of the true conditional posterior. We refer to this as *Algorithm 2*; the following is a summary of

the procedure to form a new sample (θ_{t+1}, S_{t+1}) from the current state (θ_t, S_t) of the chain, given a set of observations O :

1. Sample $\theta^* \mid S_t$, as detailed above.
2. Sample $S^* \mid O, \theta^*$:
 - a) Choose a truncation point m^* , defining a finite state-space.
 - b) Run the FFBS algorithm to draw S^* .
3. Calculate the acceptance ratio α :
 - a) Compute $p^{(t+1)}(S^* \mid \theta^*, O)$ and $p^{(t)}(S^* \mid \theta^*, O)$, the conditional posterior probabilities of the *new* trajectory under the *new* and *old* truncations.
 - b) Compute $p^{(t+1)}(S_t \mid \theta^*, O)$ and $p^{(t)}(S_t \mid \theta^*, O)$, the conditional posterior probabilities of the *old* trajectory under the *new* and *old* truncations.
 - c) Set $\alpha = \frac{p^{(t+1)}(S^* \mid \theta^*, O)p^{(t+1)}(S_t \mid \theta^*, O)}{p^{(t)}(S_t \mid \theta^*, O)p^{(t)}(S^* \mid \theta^*, O)}$.
4. With probability $\min(\alpha, 1)$, accept the new sample and set $(\theta_{t+1}, S_{t+1}) = (\theta^*, S^*)$; otherwise, set $(\theta_{t+1}, S_{t+1}) = (\theta_t, S_t)$

Note that the analysis from Section 3.1 giving the conditional posterior of the parameters (Equation (4)) still holds and is not affected by the truncation. Step 1 is therefore performed following (4). In Step 2a, we follow the Russian Roulette methodology as in Section 3.2.2 and take m^* to be the number of terms before the truncation stops. In the scheme used in our experiments, the probability of taking an additional term follows a geometric distribution, as with the previous algorithm. Based on this truncation point m^* , we can define a state-space

$$\mathcal{S} = \{(x_1, x_2, \dots, x_M) \in \mathbb{N}^M \mid x_i \leq y_i^* + m^*\}$$

where $y^* = (y_1, \dots, y_M)$ is a vector of the maximum values observed in each dimension. The method of Section 3.1 can then be used to sample a trajectory (Step 2b) in this finite state-space.

Steps 3a and 3b involve the computation of probabilities which can be performed via the forward-backward algorithm on the appropriate state-spaces. So far in this paper, the algorithm has been used to sample a new path from the process, but it can easily be adapted to calculate the probability of a given path, as shown in the algorithm outline below.

In the following, we assume we have N observations y_i at time points t_i , $i = 1, \dots, N$. For a finite state-space \mathcal{S} , we denote with \mathcal{S}_k the k -th state of the space, according to some arbitrary order. The forward and backward messages are vectors of size $|\mathcal{S}|$, and there is one such message for each observed time point. $\mathbf{a}^{(i)}$ denotes the forward message at the i -th time point t_i ; its k -th element is

$$a_k^{(i)} = p(y_1, \dots, y_{i-1}, \mathcal{S}_k)$$

that is, the joint probability of the observations prior to t_i and the state at t_i being \mathcal{S}_k . Similarly, the backward messages $\mathbf{b}^{(i)}$ has elements:

$$b_k^{(i)} \propto p(\mathcal{S}_k \mid y_i, \dots, y_N)$$

and so the probability of the observed time-series can be computed from the $\mathbf{b}^{(i)}$. This is a slightly different than the usual formulation of the forward-backward algorithm, and necessitates

Forward-backward algorithm

Require: Observations $Y = (y_1, \dots, y_N)$, finite state-space \mathcal{S} , parameters θ

Ensure: Probability of Y

```
1: Compute transition probabilities  $p_{kl}$  between states in  $\mathcal{S}$  based on  $\theta$ 
2: for  $i = 1..N$  do
3:   Compute forward message  $\mathbf{a}^{(i)}$ 
4: end for
5: Initialise  $p \leftarrow 1$ 
6: for  $i = N..1$  do
7:   Compute backward message  $\mathbf{b}^{(i)}$ 
8:   Find index  $k$  of observation  $y_i$  in  $\mathcal{S}$ 
9:    $p = p \cdot b_k^{(i)}$ 
10: end for
11: return  $p$ 
```

the computation of the forward messages $\mathbf{a}^{(i)}$ first. The messages can be computed recursively as shown in [6].

These probabilities computed this way are then used in the acceptance ratio α (Step 4). As noted above, the acceptance step is necessary because we are not proposing trajectories from the exact conditional posterior. Instead, the truncation we impose gives an estimate of the correct proposal distribution $p(S^* \mid \theta^*, O)$, and the ratio compensates for this estimate. Note that, if we could draw trajectories from the whole state-space without truncating it, the terms in α would cancel out, giving standard Gibbs sampling with acceptance rate of 1.

It is important to observe that this auxiliary variable Gibbs sampler actually targets the joint posterior distribution of parameters and trajectories. As such, it provides richer information than the M-H sampler (which directly targets the parameter posterior), but may be less effective if one is solely interested in parameter inference. The performance can also be affected by computational factors, particularly the costs of drawing sample trajectories (which was not needed in Algorithm 1, where we compute the likelihood by matrix exponentiation). In general, such costs will be model- and data-dependent, so that some initial exploration may be advisable before deciding which algorithm to use.

4 Results

This section describes the experimental validation of our approach. The experiments were performed on MATLAB implementations of the algorithms described in the previous section². The M-H proposals for Algorithm 1 were Gaussian, with variances tuned using trial runs. In the same algorithm, matrix exponentiation was performed using the method of [9], with the code that the authors have made available. Unless otherwise noted, the Russian Roulette truncation used in the experiments was chosen so as to yield 5.6 terms on average.

4.1 Variance of the estimator

Before showing how our algorithms perform against the state of the art, we present empirical evidence that our Russian Roulette-style truncation approach produces estimators with low

²available at <https://github.com/ageorgou/roulette>

$a = 0.95$ (5.6 terms)	$a = 0.75$ (2.4 terms)	$a = 0.2$ (1.2 terms)
0.0002	0.0008	0.0223
0.0051	0.0151	0.0344
0.0003	0.0013	0.0245
$< 10^{-4}$	$< 10^{-4}$	0.0016
$< 10^{-4}$	$< 10^{-4}$	0.0008
0.0005	0.0021	0.0109
$< 10^{-4}$	$< 10^{-4}$	$< 10^{-4}$
0.0003	0.0014	0.0223
0.0002	0.0011	0.0073
0.0002	0.0009	0.0077

Table 1: Coefficient of variation for the log-likelihood, estimated from 1000 samples for the LV model, under three truncation schemes (varying α) and ten parameter configurations (Section 4.1)

variance, an issue that has recently received attention in pseudo-marginal methods [14, 15]. In order to achieve estimators of low variance, the tails of the distribution of the number of terms taken must match those of the sequence being approximated [16] (or the estimator is likely to ignore significant terms). To our knowledge, there are no established results on the behaviour of transient probabilities in general pMJP as the state-space grows. Our approach is to use a geometric truncation distribution, which is well known ([17]) to arise as a steady-state distribution of simple pMJP as queueing systems, and might thus be a plausible candidate distribution. Our focus in this section is to provide an empirical evaluation of our method. Additionally, we show that the estimator is robust to the choice of the particular stopping distribution q_n used in the truncation scheme. To verify this, we considered three different q_n sequences, applied to the predator-prey model described in Section 4.2. For clarity, we write $\bar{q}_n \equiv 1 - q_n$, the probability of continuing at term n . All schemes were of the form $\bar{q}_n = a\bar{q}_{n-1}$ with $\bar{q}_0 = 1$ and $a \in \{0.95, 0.75, 0.2\}$, respectively yielding 5.6, 2.4 and 1.2 terms on average. For each scheme, we calculated 1000 estimates of the transition probabilities between observations, obtaining estimates of the log-likelihood and computing its mean and variance. This was repeated for 10 different parameterizations of the model. It can be seen (Table 1) that the variance of the estimator (measured as the coefficient of variation of the log-likelihood, due to small values) is consistently low. This validates our approach and indicates that the stopping distribution does not critically affect performance and therefore does not require fine-tuning.

An intuitive explanation for this comes from remembering that the “base” space (corresponding to the first term in the expansion) comprises all states between consecutive observations. Often, this is large enough that there is a substantial probability of the process remaining within or around it. Hence, even with a few terms, we are capturing a large part of the probability mass, and obtaining good estimates. We expect our estimator to have low variance if the process does not change radically between the observation times. It is possible, however, to find situations where the truncation strategy needs many terms in order to yield good performance. This is more likely to occur if the process is very sparsely observed, or if it is highly volatile. In both cases, the observations may not be very indicative about the behaviour of the process during the interval under consideration, therefore only taking few terms may produce inaccurate estimates of the true likelihood. This situation is also likelier when high counts are involved, in

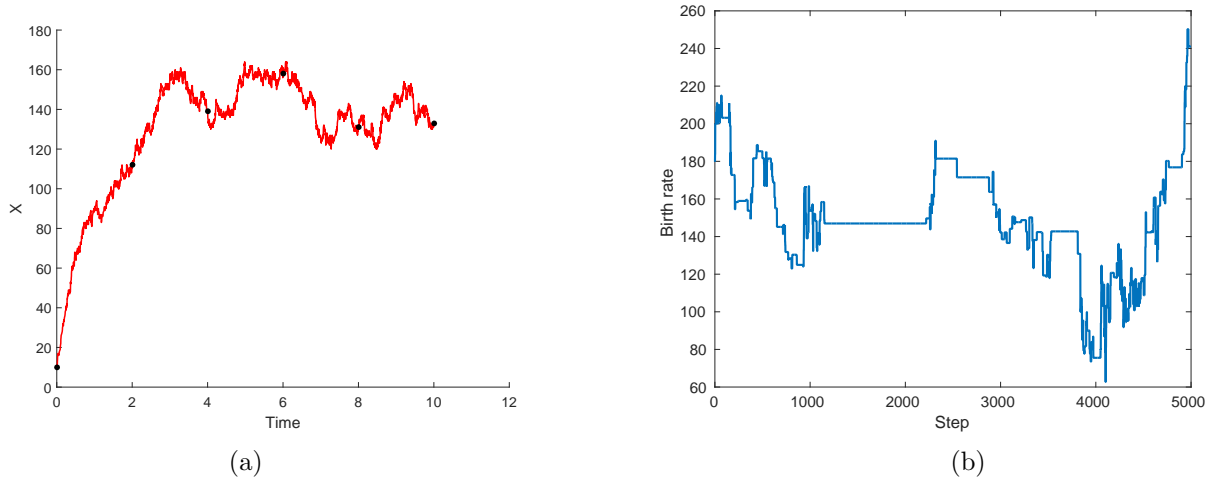


Figure 2: (a) Full trace (continuous line) and observations (dots) used in the birth-death process example; (b) Parameter samples for the birth rate using Algorithm 1, illustrating undesirable “sticking” behaviour when taking 5.6 terms on average

which case other proposed solutions are more appropriate (discussed in Section 5).

To illustrate this, we considered the example of a birth-death process involving a single species, X , with a constant birth rate of 150 and a death rate of X . From an initial value of $X = 10$, we simulated the system and used the values at 5 time points (Figure 2a). The three truncation schemes described above did not yield accurate estimates, even when taking 5 terms on average. With a stopping scheme $\bar{q}_n = 0.99\bar{q}_{n-1}$ (corresponding to 12.5 terms on average), we were able to get good estimates of the true probabilities. The more aggressive truncation schemes display higher variance and could cause problems when their estimates are used in Algorithm 1: when taking 5.6 terms on average, the variance causes the sampling chain to “stick”, as seen in Figure 2b.

As a way of improving the behaviour of the sampler, we examined the use of the so-called Monte Carlo within Metropolis (MCWM) pseudomarginal variant [13], in which the estimate of the likelihood of the current state of the chain is recomputed at every step. This can potentially alleviate the “sticking” problem and lead to better mixing, but at the cost of making the resulting chain sample from an approximation instead of the true posterior. Experiments on the predator-prey model of Section 4.2 showed that there was no noticeable improvement in either the number of steps needed to reach convergence or the acceptance rate when using MCWM. This, in addition to the bias introduced and the additional computational burden from re-estimating the likelihood, leads us to believe that in this case there is no benefit from using MCWM.

To further study of the impact of the choice of truncation distribution, we examined how it affects convergence. We tried ten different stopping distributions q_n of the form described above, chosen so that they produce $1, 2, \dots, 10$ terms on average. For each of them, we measured the steps required for convergence, as described in the next section. Overall, we found that taking more terms generally leads to faster convergence (Figure 3). This indicates that the variance of the estimates decreases when taking more terms.

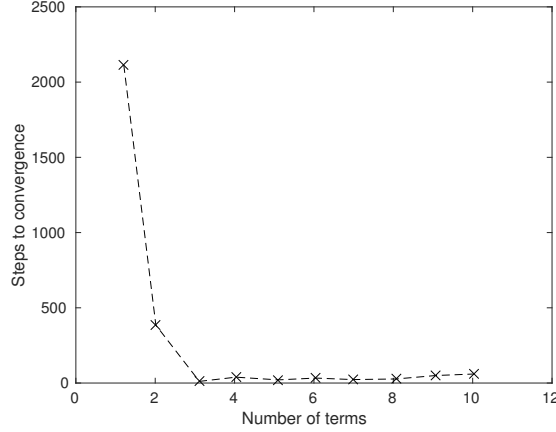
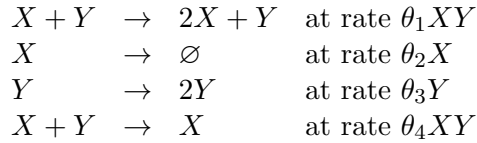


Figure 3: Steps until convergence for different stopping distributions (results shown for one parameter of the LV model).

4.2 Benchmark data sets

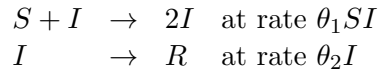
We now assess the performance the two algorithms described in the previous section as well as the Gibbs sampler based on uniformisation (Section 3.1). We could not run the original Gibbs sampler of [6] as the high number of parameters (one per state) swiftly led to storage problems. We first compared the performance of the three methods on two widely used pMJP models:

Lotka-Volterra (LV) model This predator-prey system involves four types of reactions, representing the birth and death of each species, and is a classic model in ecology and biochemistry. Truncated LV processes have been studied in previous work ([2],[18]), making it an attractive candidate for evaluating our approach.



We start from an initial state of 7 predators and 20 prey. When a finite state-space is required, we impose a maximum count of 100 for each species, as in previous work.

SIR epidemic model A commonly-used model of disease spreading (see e.g. [19]), where the state comprises three kinds of individuals: S(usceptible), I(nfected) and R(ecovered). We examine two variants of the model, a finite version where the total population is constant:



and an infinite state variant where new individuals can join the S population with unknown arrival rate:



The initial state in both cases is $(S, I, R) = (10, 5, 0)$. For the finite-state version, this gives a state-space of 121 states. For the infinite case, we chose a truncation with upper limit $(28, 33, 33)$, corresponding to 18 new arrivals in the system. To see this, note that the number

		Gibbs	Alg.1	Alg. 2
LV	Time	1011min	55min	29min
	Error	14%	10.66%	12.75%
	ESS/min	0.63	0.67	4.5
	Iter.	24	1314	180
SIR finite	Time	1min	10min	4min
	Error	2.24%	13.17%	2.13%
	ESS/min	1752.13	63.01	422.72
	Iter.	13	33	27
SIR infinite	Time	1585min	291min	666min
	Error	31.6%	25%	24.3%
	ESS/min	0.45	2.6	0.23
	Iter.	5	65	136

Table 2: Performance of the various algorithms tested. Metrics are averaged over all parameters. Experiments were performed on a 24-core Xeon E5-2680 2.5GHz, to accommodate the increased memory requirements of some cases.

of arrivals in a time interval of duration T is Poisson-distributed, with mean $\theta_3 T$. We used the final observation time and the prior mean of θ_3 , and chose the 95-percentile of the distribution governing the new arrivals. In broad terms, this means our truncation will accommodate new arrivals with 95% probability.

Table 2 summarises our evaluation results across the models considered; the metrics we use are total computational time for 5000 samples, mean relative error in parameter estimates (using the posterior mean as a point estimate), Effective Sample Size (ESS) per minute of computation, and number of iterations to convergence, defined as Potential Scale Reduction Factor (PSRF) < 1.1 [20].

Results on the LV model show that methods based on random truncations achieve very considerable improvements in performance compared to the Gibbs sampler (where the state space was truncated at a maximum number of 100 individuals per species). In particular, Algorithm 2 shows excellent behaviour in most aspects, with a high ESS suggesting it is a more efficient sampler. The running time of Algorithm 2 is comparable to that reported for a variational mean field approximation in [2], and its rapid convergence time suggests that this is a very competitive algorithm in practice. Sample results from Algorithm 2 are presented in Figure 4a for the reaction parameters, and in Figure 4b for the state of the process itself. Algorithm 1, while still computationally feasible, requires a long time to converge, reflecting potential difficulties in choosing effective proposal distributions (a problem naturally bypassed by Algorithm 2). The simple Gibbs algorithm is much slower than the other two, undoubtedly owing to its large state-space of 10000 states and very high memory requirements during the FFBS algorithm. Note that the impact of the (necessarily) large truncation is twofold. Firstly, the large state-space directly affects the running time of the FFBS algorithm, whose complexity is quadratic in the number of states. Secondly, since the rates in this model are increasing functions, having states with high counts means the generator matrix has high diagonal entries (exit rates). This, in turn, requires choosing a high exit rate for uniformisation, leading to long paths with many self-jumps, and ultimately further slowing down the FFBS step. The results for this model clearly show the usefulness of the random truncation approach compared to using

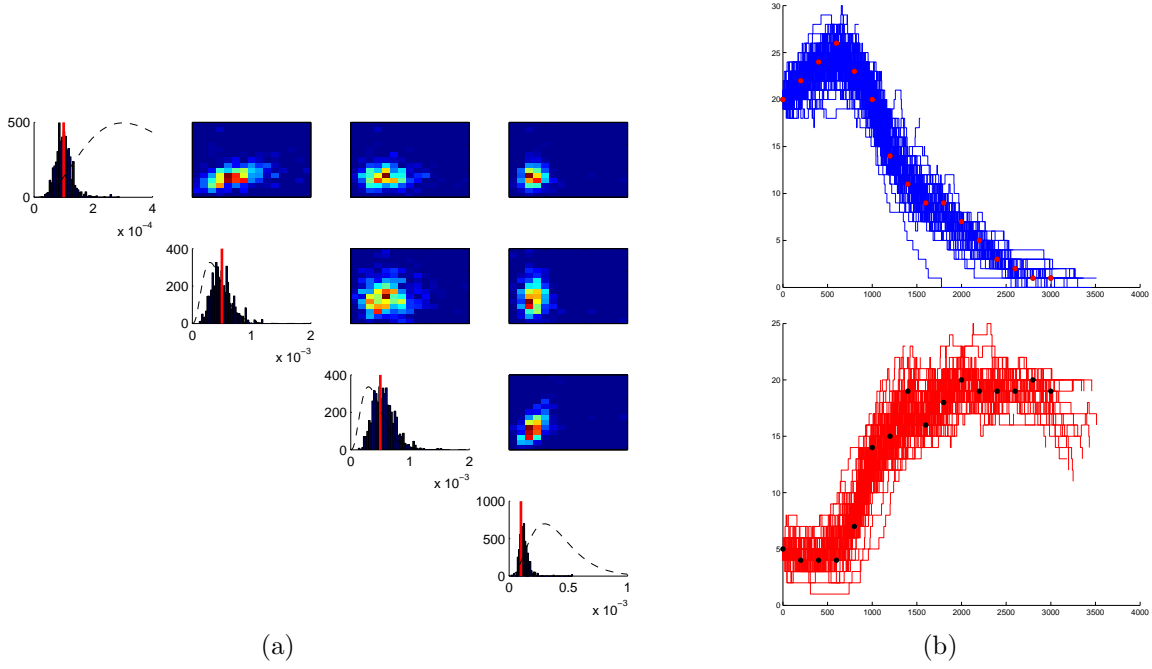


Figure 4: (a) Posterior marginals and pairwise correlations for the parameters of the LV model, from 5000 samples using Algorithm 2 (true values marked by red line, prior shown in dashed line); (b) Samples of the posterior process: prey (top), predators (bottom). Dots indicate the observations.

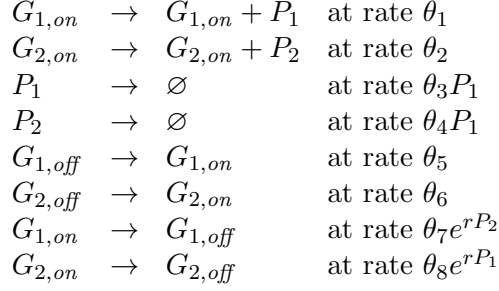
a static, conservative truncation.

Results on the SIR model show that, in the finite state space case, the Gibbs sampler of Section 3.1 is highly efficient and by some way the best algorithm. This is unsurprising, as truncations incur additional computational overheads which are not needed for such a small state space. The picture is completely different for the infinite SIR model. In this case, the M-H sampler clearly seems to be the best algorithm, achieving very fast convergence and outperforming the other two. For parameter values within the prior range, the infinite SIR model exhibits fast dynamics which lead to very long uniformised trajectories, considerably increasing the computational costs of sampling trajectories via the FFBS algorithm. The problem is further compounded for the simple Gibbs sampler algorithm of Section 3.1. Even with the truncation described above, there are 32594 states, resulting in very severe computational and storage costs.

4.3 Genetic toggle switch

As a real application of our approach, we consider a model of a synthetic biological circuit describing the interaction between two genes (G1 and G2) and the proteins they encode (P1 and P2). Each protein acts as a repressor for the other gene, inhibiting its expression. This leads to a bistable behaviour, switching between a state with high P1 and low P2, and one with low P1 and high P2 (hence the name *toggle-switch*). The interactions are encoded as eight

chemical reactions:



where r is a constant assumed known.

This system was engineered *in vivo* in one of the pioneering studies in synthetic biology [21] and has been further studied in [22]. Statistical inference is increasingly being recognised as a crucial bottleneck in synthetic biology: while genome engineering technologies enable researchers to reliably synthesise circuits with a desired structure, predicting the dynamic behaviour of a circuit requires knowledge of the kinetic parameters of the system once it is implanted *in the cell*, which cannot be directly measured. As synthetic biology is intrinsically at the single cell level, inference techniques for stochastic models have the potential to be of great aid in the rational design of synthetic biology circuits.

Following [22], we model the system using a binary state for each gene and discrete levels for the proteins. The genes can be active or inactive, with protein being produced only in the former case. Each gene can be modelled with a telegraph process: an inactive gene becomes active at a constant rate, and an active one becomes inactive at a rate depending on the level of its repressor. When a gene is active, the level of its product follows a birth-death process; that is, proteins are produced at a constant rate and degrade at mass-action rates. We use a single production reaction for each protein to abstract various underlying mechanisms, including transcription and translation. The model comprises eight types of reaction; note that the requirements of our method on the form of the kinetic laws (Section 3.1) are flexible enough to accommodate the deactivation dynamics used here, even though they are not mass-action. We simulated the system to produce behaviour similar to the simulated traces in [22]. We kept 20 time points of measurements, which varied between 0 and 24 for each observed protein.

We used Algorithm 2 to infer the joint posterior distribution of the eight parameters and state trajectories in this system. Our results indicate that the likelihood is relatively insensitive to the parameters governing the activation and deactivation of the two genes. This is a reasonable result, since we do not observe the state of the genes but only the levels of the two protein products. Therefore, the effect of the switching parameters is seen only indirectly through the switching events, which are rare in the data. In contrast, the protein expression and degradation rates have sharp posteriors which capture interesting correlations between the parameters — for instance, we observe a strong correlation between the production and degradation rate of each protein, as perhaps expected given the similarity to a birth-death process. Figure 5 shows parameter posteriors and convergence statistics for one such experiment, showcasing the good behaviour of the algorithm.

5 Related work

Parameter inference in pMJPs has been the subject of previous work, with a significant body of literature focusing on continuous approximations to the process, in order to work around the complexities entailed by the stochastic dynamics. In general, such approximations are

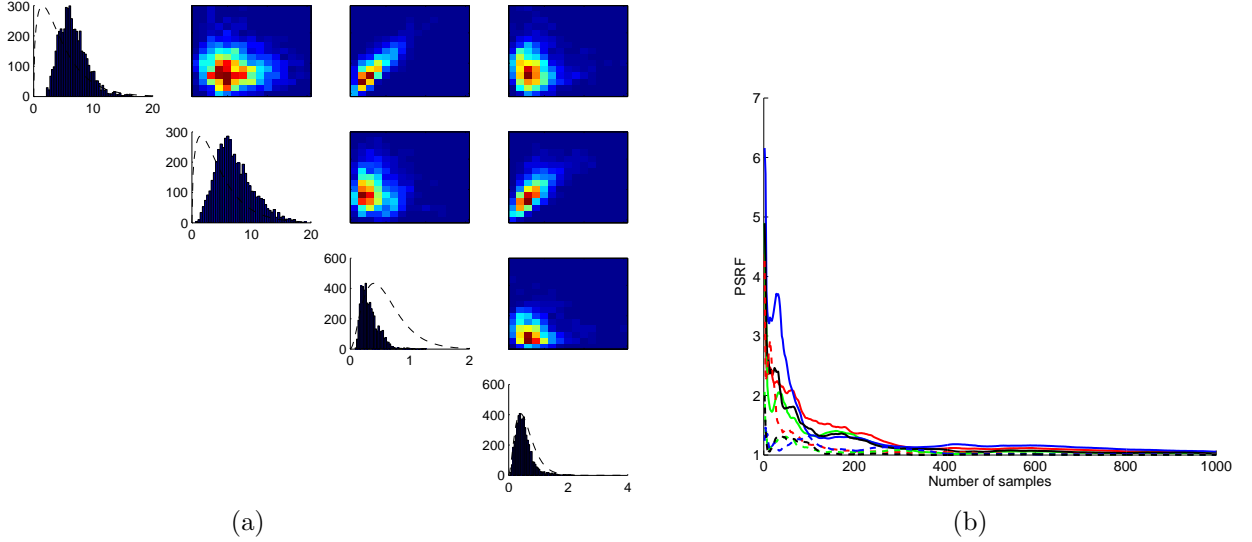


Figure 5: (a) Posterior marginals and pairwise correlations for four parameters ($\theta_1, \theta_2, \theta_3$ and θ_4) of the toggle switch model, from 5000 samples using Algorithm 2 (priors shown in dashed line); (b) PSRF for all eight parameters.

more accurate when the populations involved are high, and their accuracy degrades for lower populations as the impact of discrete stochastic behaviour becomes more pronounced. Two general classes of methods have been proposed to this end. The first involves approximating a pMJP with a diffusion process, as in [23], and using the resulting stochastic differential equations to calculate the likelihood. The second approach uses van Kampen’s Linear Noise Approximation [24], which assumes that the marginal distribution of the approximating process at any time is Gaussian. Under this assumption, ordinary differential equations for the mean and covariance can be derived as in [25, 26, 27] and used to compute the likelihood as part of an inference scheme. In contrast to these methods, our suggested approach is expected to be more accurate for smaller populations, as it maintains the stochastic dynamics. This makes it particularly useful for a range of systems which are large enough that a direct solution is inefficient, but not as large as to be accurately represented with continuous dynamics.

In addition to MCMC-based approaches, like ours, particle methods have also been proposed for use with pMJPs, either with the exact dynamics [5, 4] or with continuous approximations such as the ones mentioned above. However, they do require more user choices (e.g. number of particles) and can also incur heavy computational overheads for large models or state spaces. For infinite MJPs, in particular, the transition kernel is not available explicitly, making particle methods non-trivial and intrinsically expensive. Variational methods have been developed in [2], and can offer computational savings; however, the work in [2] only performed state inference, providing point estimates for parameters. Furthermore, the error introduced by the variational approximation is often difficult to quantify.

Recent work has made use of random truncations in different contexts: Strathmann *et al.* [28] propose using a Russian Roulette-style approach in large data scenarios where computing the likelihood from all data points is impractical, while Filippone & Engler [8] exploit the methodology to perform efficient inference for Gaussian processes. More generally, the construction of unbiased estimators has been the subject of theoretical and practical analysis. McLeish [29] and Rhee & Glynn [16] examine the use of a method similar to Russian Roulette for obtaining un-

biased estimates from biased ones. Agapiou *et al.* [30] consider ways of debiasing the estimates obtained by MCMC methods, particularly focusing on infinite spaces. Jacob & Thiery [31] examine the theoretical existence of estimators that are both unbiased and guaranteed to be non-negative under different generation schemes.

6 Conclusions

MJPs are common models in many branches of science, yet they still present fundamental statistical challenges. In this paper, we have proposed a novel MCMC framework for asymptotically exact inference for pMJPs, an important class of MJPs widely used in chemistry and systems biology. We remark that, while our focus is on biological applications, models with exactly the same structure are employed in many other fields, from epidemiology to ecology to performance modelling. Our random truncations pseudo-marginal approach enables a principled treatment of systems with potentially unbounded state-spaces. Interestingly, our results show that random truncations can also bring computational benefits over the naive alternative of bounding the state-space *ab initio*, as done in [6]. Intuitively, this is because choosing a truncation which guarantees a certain error bound usually requires still retaining a large state-space, while our random truncation method generally samples from much smaller systems. The two truncation-based algorithms we consider here appear to perform best in different kinds of systems, and so neither can be said to be clearly superior in the general case.

The performance of our proposed methods may vary with the system in question. As the number of species grows, the state-space grows exponentially larger, leading to increased computational overheads for our method (as for many other methods). While this may be a serious limitation for large models, it is worth pointing out that many practical applications of pMJPs describe systems with a small number of species, where our method’s performance should not be affected. High counts of the species involved also result in larger state-spaces, leading to heavier computations, particularly for Algorithm 1. For Algorithm 2, the rates of the reactions can also have an impact: very fast reactions lead to a fine time-discretization and slower computations in the forward-backward step. Our methods perform best when particle numbers are not exceedingly large (otherwise, a continuous approximation would be both accurate and more efficient) and when observations are relatively dense or, equivalently, the process is not too volatile (or a truncation with many terms would be required for a good result).

Pseudo-marginal methods based on random truncations are relatively new to statistics and machine learning [32, 7, 8]: to our knowledge, this is the first time that they are employed as a way of truncating an unbounded state space, and we think this idea may be appealing in other scenarios where unbounded state spaces are normal, such as non-parametric Bayesian methods. Compared to pseudo-marginal methods based on importance sampling [33, 5], random truncations offer several advantages: there is no need to choose a proposal distribution, a notoriously difficult problem in high dimensions. The choice of the truncating distribution, which controls the variance of the estimator, can in general be aided by some initial exploratory runs with different truncation distributions with different expected numbers of retained terms. Recent work on improving the behaviour of pseudo-marginal MCMC methods [34] may also be relevant to enhancing the performance of our proposed method.

Acknowledgements JH acknowledges support from the EU FET-Proactive programme through QUANTICOL grant 600708. GS acknowledges support from the European Research Council through grant MLCS306999. This work was supported by Microsoft Research through its PhD

Scholarship Programme. The authors thank Maurizio Filippone, Mark Girolami and Chris Sherlock for useful discussions, Yichuan Zhang and Vinayak Rao for providing code and advice on computations, and the anonymous reviewers for their helpful feedback.

References

- [1] Crispin W Gardiner. *Handbook of Stochastic Methods for Physics, Chemistry and the Natural Sciences*, vol. 13 of. Springer Series in Synergetics, 1985.
- [2] Manfred Opper and Guido Sanguinetti. Variational inference for Markov jump processes. In J.C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 2008*, pages 1105–1112. MIT Press, Cambridge, MA, 2008.
- [3] Ido Cohn, Tal El-Hay, Nir Friedman, and Raz Kupferman. Mean Field Variational Approximation for Continuous-Time Bayesian Networks. *The Journal of Machine Learning Research*, 11:2745–2783, 2010.
- [4] Christoph Zechner, Michael Unger, Serge Pelet, Matthias Peter, and Heinz Koepl. Scalable inference of heterogeneous reaction kinetics from pooled single-cell recordings. *Nature Methods*, 11(2):197–202, 2014.
- [5] Monir Hajiaghayi, Bonnie Kirkpatrick, Liangliang Wang, and Alexandre Bouchard-Côté. Efficient Continuous-Time Markov Chain Estimation. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 638–646, 2014.
- [6] Vinayak Rao and Yee Whye Teh. Fast MCMC sampling for Markov jump processes and extensions. *Journal of Machine Learning Research*, 14:3207–3232, 2013. arXiv:1208.4818.
- [7] Anne-Marie Lyne, Mark Girolami, Yves Atchad, Heiko Strathmann, and Daniel Simpson. On Russian Roulette Estimates for Bayesian Inference with Doubly-Intractable Likelihoods. *Statist. Sci.*, 30(4):443–467, 11 2015. doi: 10.1214/15-STS523. URL <http://dx.doi.org/10.1214/15-STS523>.
- [8] Maurizio Filippone and Raphael Engler. Enabling scalable stochastic gradient-based inference for Gaussian processes by employing the Unbiased Linear System SolvEr (ULISSE). In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, July 6-11, 2015*, 2015.
- [9] Awad H. Al-Mohy and Nicholas J. Higham. Computing the Action of the Matrix Exponential, with an Application to Exponential Integrators. *SIAM Journal on Scientific Computing*, 33(2):488–511, 2011. doi: 10.1137/100788860. URL <http://dx.doi.org/10.1137/100788860>.
- [10] Arne Jensen. Markoff chains as an aid in the study of Markoff processes. *Scandinavian Actuarial Journal*, 1953(sup1):87–91, 1953.
- [11] Brian Munsky and Mustafa Khammash. The finite state projection algorithm for the solution of the chemical master equation. *The Journal of Chemical Physics*, 124(4):044104, 2006. doi: <http://dx.doi.org/10.1063/1.2145882>. URL <http://scitation.aip.org/content/aip/journal/jcp/124/4/10.1063/1.2145882>.

- [12] Christophe Andrieu and Gareth O. Roberts. The pseudo-marginal approach for efficient Monte Carlo computations. *The Annals of Statistics*, pages 697–725, 2009.
- [13] Mark A. Beaumont. Estimation of population growth or decline in genetically monitored populations. *Genetics*, 164(3):1139–1160, 2003. ISSN 0016-6731. URL <http://www.genetics.org/content/164/3/1139>.
- [14] A. Doucet, M. K. Pitt, G. Deligiannidis, and R. Kohn. Efficient implementation of Markov chain Monte Carlo when using an unbiased likelihood estimator. *Biometrika*, 2015. doi: 10.1093/biomet/asu075. URL <http://biomet.oxfordjournals.org/content/early/2015/03/07/biomet.asu075.abstract>.
- [15] Chris Sherlock, Alexandre H. Thiery, Gareth O. Roberts, and Jeffrey S. Rosenthal. On the efficiency of pseudo-marginal random walk Metropolis algorithms. *Ann. Statist.*, 43(1):238–275, 02 2015. doi: 10.1214/14-AOS1278. URL <http://dx.doi.org/10.1214/14-AOS1278>.
- [16] CH Rhee and Peter W Glynn. Unbiased estimation with square root convergence for sde models. *To appear*. URL <http://rhee.gatech.edu/papers/RheeGlynn13a.pdf>.
- [17] Leonard Kleinrock. *Queueing Systems*, volume I: Theory. Wiley Interscience, 1975.
- [18] Richard J Boys, Darren J Wilkinson, and Thomas BL Kirkwood. Bayesian inference for a discretely observed stochastic kinetic model. *Statistics and Computing*, 18(2):125–135, 2008.
- [19] Roy M Anderson and Robert McCredie May. *Infectious diseases of humans*, volume 1. Oxford university press Oxford, 1991.
- [20] Andrew Gelman, John B Carlin, Hal S Stern, and Donald B Rubin. *Bayesian Data Analysis*. Taylor & Francis, 2014.
- [21] Timothy S Gardner, Charles R Cantor, and James J Collins. Construction of a genetic toggle switch in *Escherichia coli*. *Nature*, 403(6767):339–342, 2000.
- [22] Tianhai Tian and Kevin Burrage. Stochastic models for regulatory networks of the genetic toggle switch. *Proceedings of the National Academy of Sciences*, 103(22):8372–8377, 2006. doi: 10.1073/pnas.0507818103.
- [23] A. Golightly and D. J. Wilkinson. Bayesian Inference for Stochastic Kinetic Models Using a Diffusion Approximation. *Biometrics*, 61(3):781–788, 2005. ISSN 1541-0420. doi: 10.1111/j.1541-0420.2005.00345.x. URL <http://dx.doi.org/10.1111/j.1541-0420.2005.00345.x>.
- [24] Nicolaas Godfried Van Kampen. *Stochastic processes in physics and chemistry*, volume 1. Elsevier, 1992.
- [25] Andreas Ruttor and Manfred Opper. Efficient Statistical Inference for Stochastic Reaction Processes. *Physical review letters*, 103(23), 2009. doi: 10.1103/PhysRevLett.103.230601. URL <http://link.aps.org/doi/10.1103/PhysRevLett.103.230601>.
- [26] Paul Fearnhead, Vasilios Giagos, and Chris Sherlock. Inference for reaction networks using the linear noise approximation. *Biometrics*, 70(2):457–466, 2014. ISSN 1541-0420. doi: 10.1111/biom.12152. URL <http://dx.doi.org/10.1111/biom.12152>.

- [27] Michał Komorowski, Bärbel Finkenstädt, Claire Harper, and David Rand. Bayesian inference of biochemical kinetic parameters using the linear noise approximation. *BMC bioinformatics*, 10(1):343, 2009. doi: 10.1186/1471-2105-10-343.
- [28] Heiko Strathmann, Dino Sejdinovic, and Mark Girolami. Unbiased bayes for big data: Paths of partial posteriors. *arXiv preprint arXiv:1501.03326*, 2015.
- [29] Don McLeish. A general method for debiasing a monte carlo estimator. *Monte Carlo Methods and Applications*, 2011.
- [30] Sergios Agapiou, Gareth O Roberts, and Sebastian J Vollmer. Unbiased monte carlo: posterior estimation for intractable/infinite-dimensional models. *arXiv preprint arXiv:1411.7713*, 2014.
- [31] Pierre E. Jacob and Alexandre H. Thiery. On nonnegative unbiased estimators. *Ann. Statist.*, 43(2):769–784, 04 2015. doi: 10.1214/15-AOS1311. URL <http://dx.doi.org/10.1214/15-AOS1311>.
- [32] Chang-han Rhee and Peter W. Glynn. A new approach to unbiased estimation for sde’s. In *Proceedings of the Winter Simulation Conference*, WSC ’12, pages 17:1–17:7, 2012. URL <http://dl.acm.org/citation.cfm?id=2429759.2429780>.
- [33] Maurizio Filippone and Mark A. Girolami. Pseudo-Marginal Bayesian Inference for Gaussian Processes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 36(11):2214–2226, 2014. doi: 10.1109/TPAMI.2014.2316530. URL <http://doi.ieeecomputersociety.org/10.1109/TPAMI.2014.2316530>.
- [34] Iain Murray and Matthew M. Graham. Pseudo-marginal slice sampling. *AISTATS 2016 (to appear)*, *arXiv preprint arXiv:1510.02958*.